

Amendments to the Claims

This listing of claims will replace all prior versions.

Listing of Claims

1. (Currently Amended) A method, comprising:
 - providing an interface for communication between a set of first programs and a second program during program runtime; and
 - providing to the second program at least one of a set of third programs associated with at least one of the set of first programs, in response to a dataset associated with said at least one of the set of first programs, wherein the at least one of the set of third programs selectively modifies the interface during program runtime for communication between the second program and said at least one of the set of first programs.
2. (Original) The method of Claim 1, wherein providing to the second program includes:
 - loading in the second program by one of the set of first programs; and
 - loading in at least one of the set of third programs by the second program for tuning the response of said second program to the at least one of the set of first programs.
3. (Original) The method of Claim 1, wherein providing to the second program includes loading in a fourth program by the second program for serving the at least one of the set of first programs.

4. (Original) The method of Claim 1, wherein the set of first programs includes a set of application programs for electronic design automation.
5. (Original) The method of Claim 1, wherein the second program includes a shared object having a generic code for use with the set of first programs.
6. (Original) The method of Claim 1, wherein the second program includes a dynamic link library having a plurality of generic macros for use with the set of first programs.
7. (Original) The method of Claim 1, wherein the set of third programs includes a plurality of application specific shared objects, each application specific shared object having one or more application specific macros associated with the at least one of set of first programs.
8. (Original) The method of Claim 1, wherein the set of third programs includes a plurality of application specific dynamic link libraries, each application specific dynamic link library having one or more application specific macros associated with one or more of set of first programs.
9. (Original) The method of Claim 3, wherein the fourth program includes one or more active models, each active model having a dataset and an algorithmic content, the forth program being shared by the set of first programs.

10. (Original) The method of Claim 9, wherein the at least one of the set of first programs communicates with the fourth program through the second program while utilizing the at least one of the set of third programs.

11. (Original) The method of Claim 9, wherein the at least one of the set of first programs communicates directly with the fourth program while utilizing the at least one of the set of third programs.

12. (Currently Amended) The method of Claim 1, wherein providing includes:
making a runtime call having the dataset from the at least one of the set of first programs to the second program; and
directing the runtime call to a selected one of the set of third programs responsive to a first determination from the dataset.

13. (Original) The method of Claim 12, wherein the dataset includes a first set of one or more monitored parameters and a second set of one or more operational parameters.

14. (Original) The method of Claim 13, wherein the first determination includes checking the dataset for at least one monitored parameter from the first set of one or more monitored parameters.

15. (Original) The method of Claim 14, wherein checking includes:
 - performing a first set of actions responsive to presence of the at least one monitored parameter; and
 - performing a second set of actions responsive to absence of the at least one monitored parameter.
16. (Original) The method of Claim 15, wherein performing the first set of actions includes responding to the at least one of the first set of first programs with a query for determining a next action.
17. (Original) The method of Claim 15, wherein performing the second set of actions includes optimizing a sequence of calls as a function of the dataset associated with the at least one of the first set of first programs.
18. (Original) The method of Claim 12, wherein providing includes making a callback from the at least one of the set of third programs to the at least one of the set of first programs for determining a response to the call.
19. (Original) The method of Claim 1, wherein providing includes:
 - making a call having the dataset from the at least one of the set of first programs to the second program; and
 - responding to the at least one of the set of first programs responsive to a second determination from the dataset.

20. (Currently Amended) A method for using a set of first programs with a second program during program runtime, comprising:

providing an application procedural interface for communication between the set of first programs and the second program; and

providing, through the use of the application procedural interface during program runtime, to the second program at least one of a set of plug-ins from a database responsive to a dataset identified to be associated with said at least one of the set of first programs.

21. (Original) The method of Claim 20, wherein providing includes identifying said at least one of the set of first programs to the second program by analyzing the dataset with the second program.

22. (Original) The method of Claim 20, further comprising, before providing said application procedural interface, creating said at least one of a set of the plug-ins for supporting operation of the second program with said at least one of the set of first programs.

23. (Original) The method of Claim 20, wherein the second program includes an active dynamic library including one or more active models, each of said one or more active models having an associated data and algorithmic content.

24. (Original) The method of Claim 20, wherein the set of first programs includes a plurality of application programs deployed in a design flow of an integrated circuit.

25. (Canceled)

26. (Canceled)

27. (Original) The method of Claim 25, wherein each of said second set of functions includes a callback.

28. (Currently Amended) A system, comprising:

an interface to communicate between a set of first programs and a second program during program runtime; and

a set of third programs, wherein one of the set of first programs loads in the second program during program runtime, and the second program, responsive to a dataset from one of the set of first programs, loads in at least one of the set of third programs during program runtime.

29. (Original) The system of Claim 28, wherein said dataset is identified to be associated with said at least one of the set of first programs.

30. (Original) The system of Claim 28, wherein said at least one of the set of third programs is a plug-in to said second program.

31. (Currently Amended) A system for using a set of first programs with a second program, comprising:

an application procedural interface for communication between the set of first programs and the second program during program runtime; and

a database including a set of plug-ins, wherein one of the set of first programs loads in the second program during program runtime, and the second program is responsive to a dataset from one of the set of first programs to load in at least one of the set of plug-ins during program runtime.

32. (Original) The system of Claim 31, wherein said database includes a directory having the set of plug-ins organized in a file system.

33. (Original) The system of Claim 31, wherein each of said set of plug-ins includes an application personality profile for an associated one of the set of first programs, the application personality profile determines an optimized sequence of function calls between the associated one of the set of first programs and the second program, said optimized sequence responsive to the dataset.

34. (Currently Amended) A system, comprising:

an application procedural interface for extending a dynamic library for use with a first application program and a second application program during program runtime;

a first plug-in, wherein the dynamic library loads the first plug-in responsive to the first application program during program runtime; and

a second plug-in, wherein the dynamic library loads the second plug-in responsive to the second application program during program runtime.

35. (Original) The system of Claim 34, wherein said first plug-in is stored in a library.

36. (Original) The system of Claim 35, wherein said first and second plug-ins are stored in said library.

37. (Original) The system of Claim 34, wherein said each plug-in includes:

a first set of one or more parameters to be monitored;

a first rule for at least one of the first set of one or more parameters;

a second set of one or more parameters to be processed;

a second rule for at least one of the second set of one or more parameters;

a first runtime responsive to a set of transactions through the application procedural interface, the first runtime stores information on transactions affecting one or more of the first set of one or more parameters and one or more of the second set of one or more parameters; and

a second runtime responsive to the first runtime, the second runtime invokes one of a first set of actions in response to said at least one of the first set of one or more parameters failing to comply with the first rule, and invokes one of a second set of actions in response to said at least one of the second set of one or more parameters being generated according to the second rule.

38. (Original) An electronic media, comprising a program for performing the method of Claim 1.

39. (Currently Amended) A computer program, comprising computer or machine-readable program elements ~~translatable~~ for implementing the method of Claim 1.

40. (Original) The method of Claim 1, further comprising verifying a design of an integrated circuit.

41. (Original) An integrated circuit designed in accordance with the method of Claim 1.

42. (Currently Amended) A computer program comprising computer program means adapted to perform the steps of providing an interface for communication between a set of first programs and a second program during program runtime; and providing to the second program at least one of a set of third programs associated with at least one of the set of first programs responsive to a dataset identified to be associated with said at least one of the set of first programs when said at least one of the set of first programs is run on a computer during program runtime.

43. (Original) A computer program as claimed in Claim 42, embodied on a computer-readable medium.

44. (Original) An electronic media, comprising a program for performing the method of Claim 20.

45. (Currently Amended) A computer program, comprising computer or machine-readable program elements ~~translatable~~ for implementing the method of Claim 20.

46. (Original) The method of Claim 20, further comprising verifying a design of an integrated circuit.

47. (Original) An integrated circuit designed in accordance with the method of Claim 20.

48. (Currently Amended) A computer program comprising computer program means ~~adapted~~ to perform the steps of providing an application procedural interface for communication between a set of first programs and a second program during program runtime; and providing, through the use of the application procedural interface, to the second program at least one of a set of plug-ins from a database responsive to a dataset identified to be associated with said at least one of the set of first programs when said at least one of the set of first programs is run on a computer.

49. (Original) A computer program as claimed in Claim 48, embodied on a computer-readable medium.

50. (Currently Amended) A method for using a first program with a second program, comprising:

communicating an indication from the first program to the second program;

analyzing the indication to determine an interaction between the first and second program; and

utilizing a third program to tune the interaction between the first program and the second program during program runtime.